



Independent Study: SDE project

Oral Presentation

Presentor: Selena

rg4012@nyu.edu

My Best Mentor: Prof. Loucas Pillaud-Vivien

12/29/2023



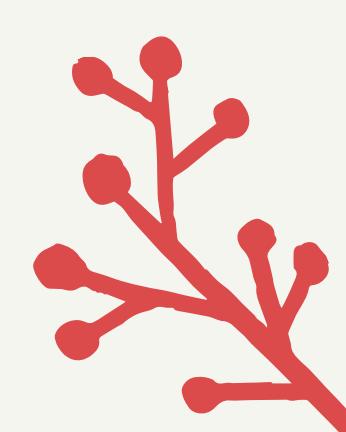
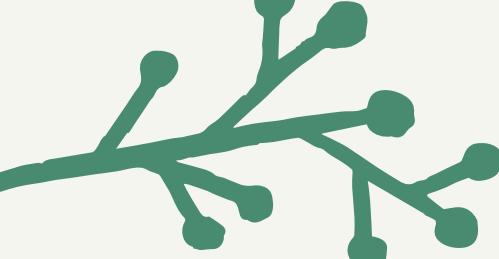
Contents:

- * Optimization Methods for Large Scale Machine Learning
- * Introduction to Stochastic Differential Equations(SDE)
- * Stochastic Gradient Descent(SGD)

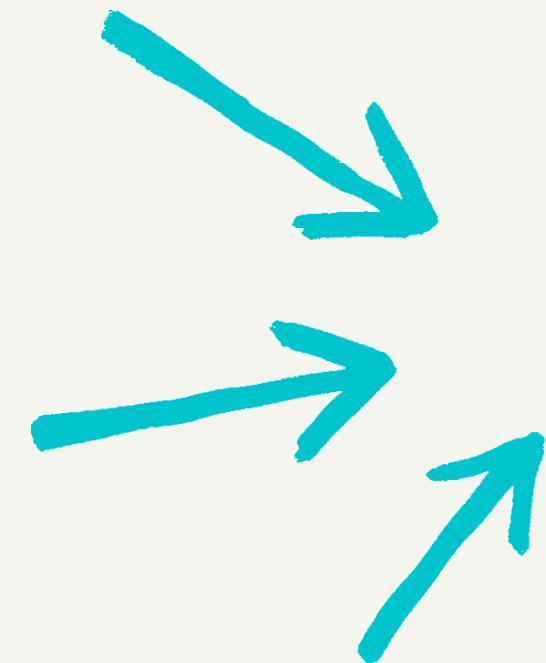
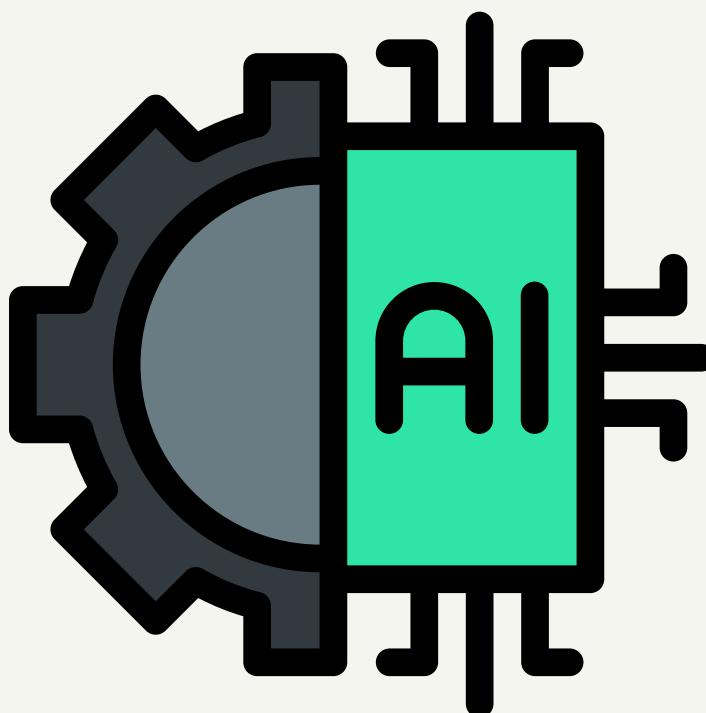
-1-

Optimization Methods for Large Scale Machine Learning





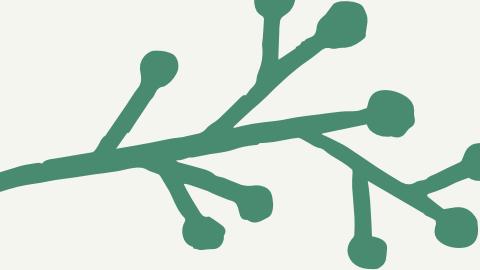
What is the pillar of
Machine Learning?



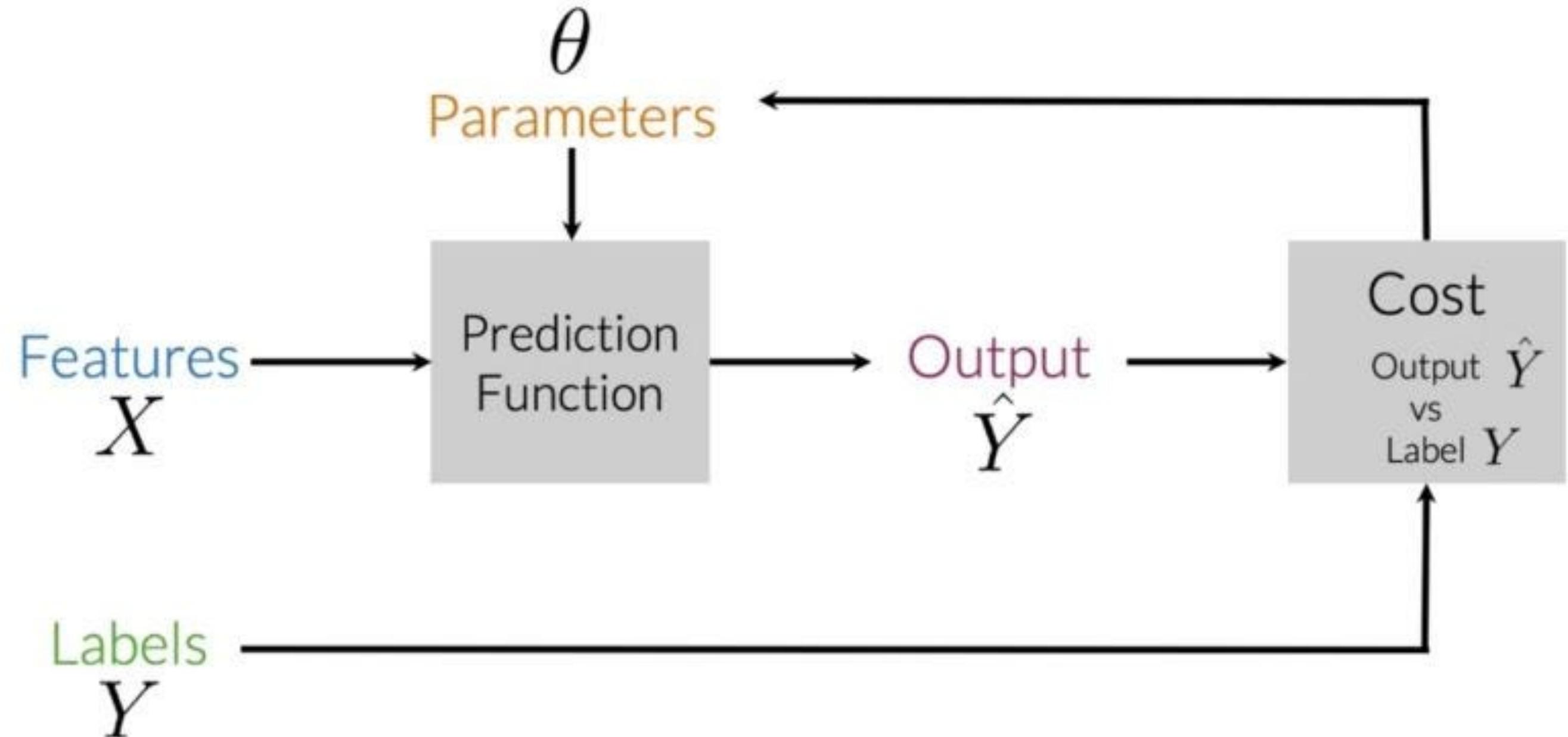
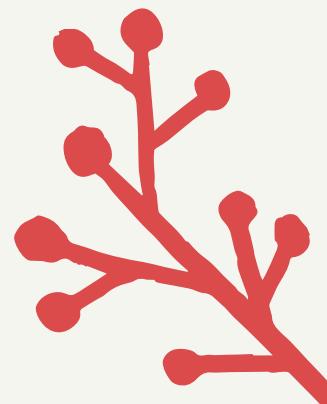
Mathematical Optimization!

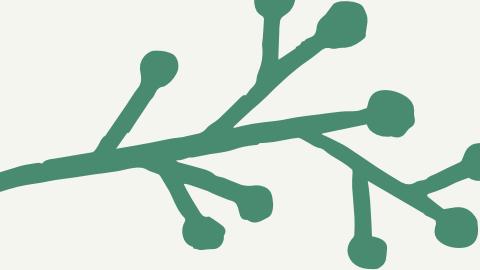
— Numerical computation for a system designed
to make decisions based on yet unseen data



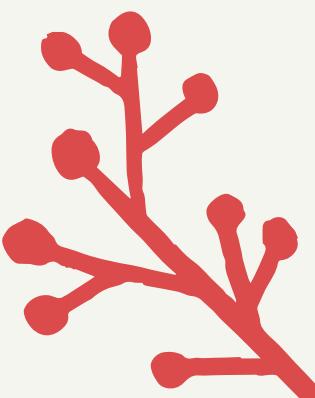


Prediction Function h





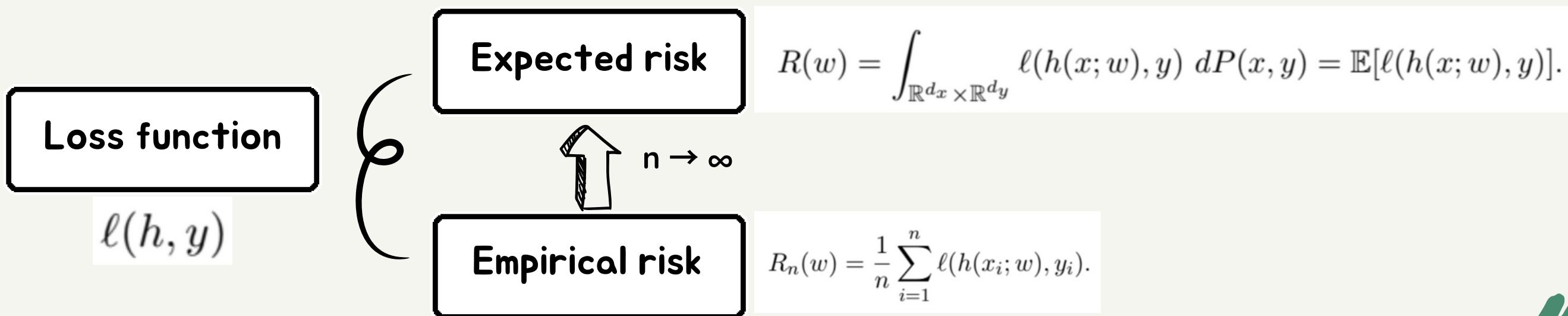
Choice of Prediction Function Family



Choice of prediction function family H :

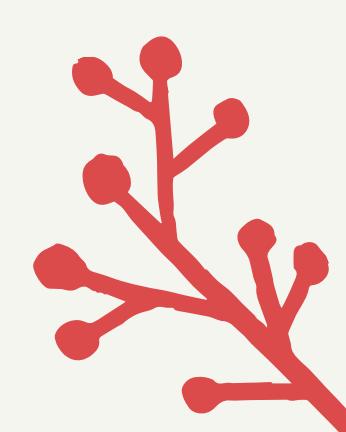
- H should contain prediction functions that can achieve a low empirical risk over the training set – Achieved by selecting a rich family of functions.
- The gap between expected risk and empirical risk, i.e. $R(h) - R_n(h)$ should be small over h in H – Achieved by uniform Law of Large Number.
- Should solve the corresponding optimization problem

$$\sup_{h \in \mathcal{H}} |R(h) - R_n(h)| \leq \mathcal{O} \left(\sqrt{\frac{1}{2n} \log \left(\frac{2}{\eta} \right)} + \frac{d_H}{n} \log \left(\frac{n}{d_H} \right) \right)$$



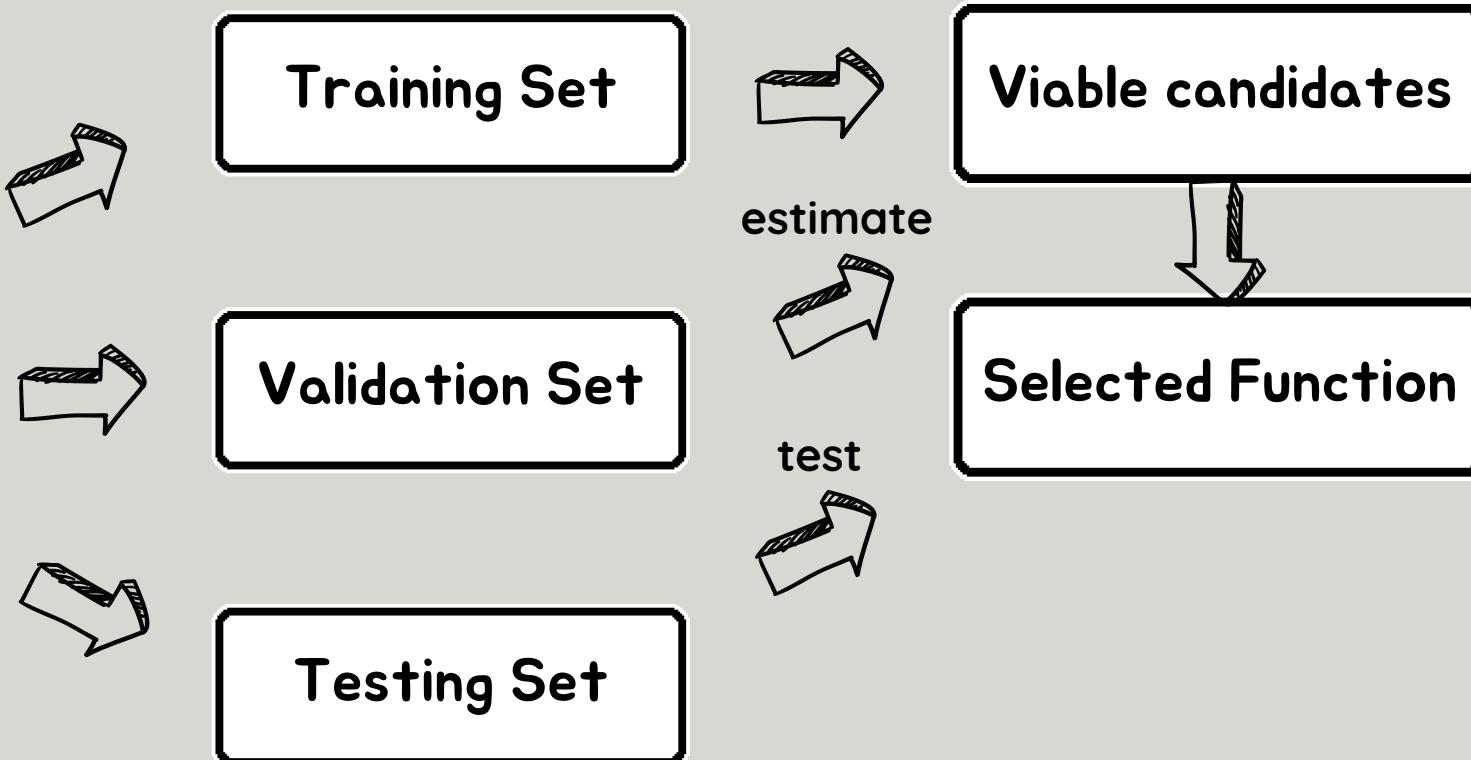


How to choose an optimal model?



Cross Validation Procedure

Mathematical Optimization:
Minimize Empirical Risk of the
prediction function h



Regularization

SRM(Structural risk minimization)

$$\min_{(w,\tau) \in \mathbb{R}^d \times \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \ell(h(x_i; w, \tau), y_i) + \frac{\lambda}{2} \|w\|_2^2.$$

ERM

(Empirical risk minimization)

Regularizer

(convex)

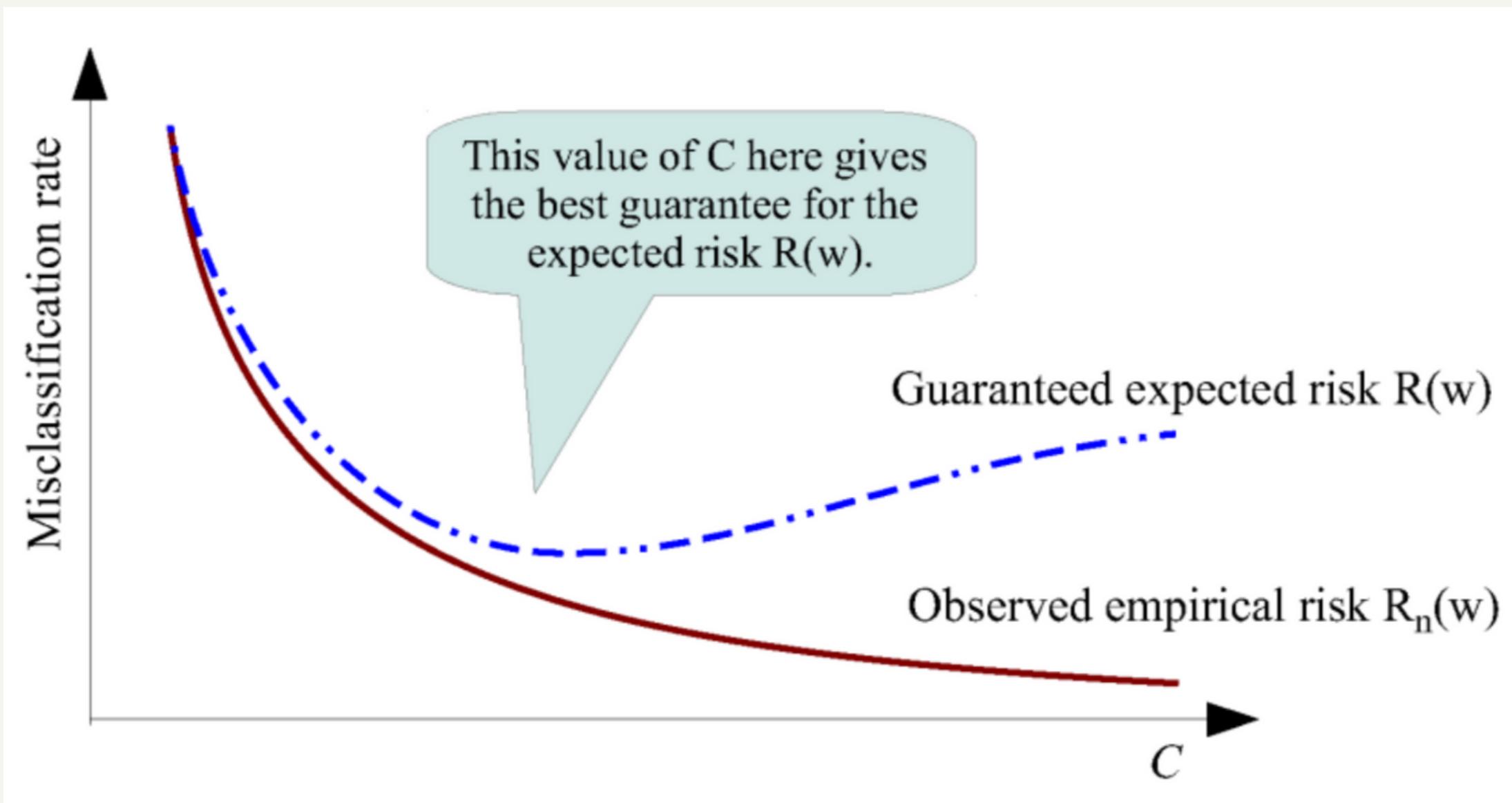


Add regularizer to avoid overfitting

Optimization method of solving ERM:
Batch vs. Stochastic Gradient Descent

Overfitting:

If the predictive ability of the training data is pursued too much, the complexity of the selected model is often higher than that of the real model, which is called overfitting. It refers to the phenomenon that the selected model contains too many parameters during learning, resulting in the model predicting well for known data and poorly for unknown data.



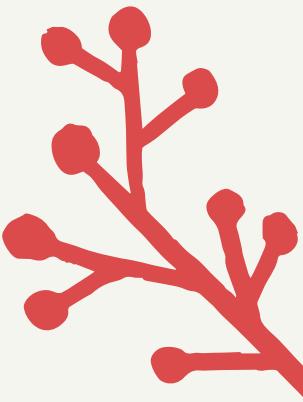
-2-

Introduction to Stochastic Differential Equations(SDE)





What is SDE?



Stochastic differential equations (SDEs) are a type of differential equations used to model systems that exhibit random behavior. An SDE typically takes the form:

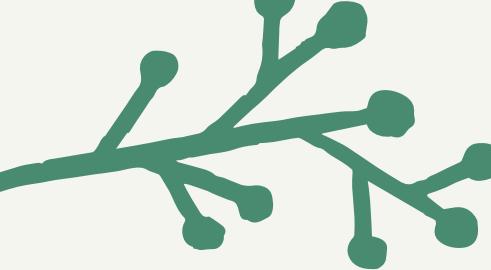
$$dX = a(t, X)dt + b(t, X)dW_t$$

- ▶ $a(t, X)dt$: drift term because it captures the average or expected rate of change of the process X if no randomness was involved.
- ▶ $b(t, X)dW_t$: diffusion term because it scales the magnitude of the randomness by the increment of W .

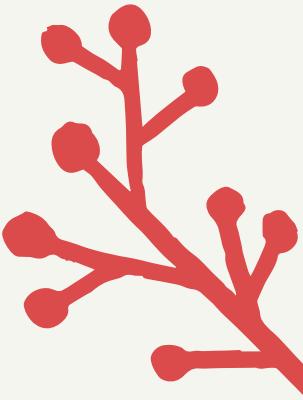
Numerical Solution of SDE:

- **Ito's formula**
(Chain rule for SDE)
- **Euler-Maruyama method**
(Approximate solution of SDE)
Typical model: OU process





Method 1: Ito's formula



Ito's formula

If $Y = f(t, X)$, then:

$$dY = \frac{\partial f}{\partial t}(t, X)dt + \frac{\partial f}{\partial x}(t, X)dx + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(t, X)dx dx$$

where the $dx dx$ term is interpreted by using the identities:

$$dt dt = 0$$

$$dt dW_t = dW_t dt = 0$$

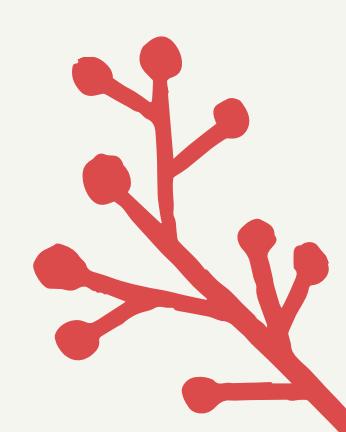
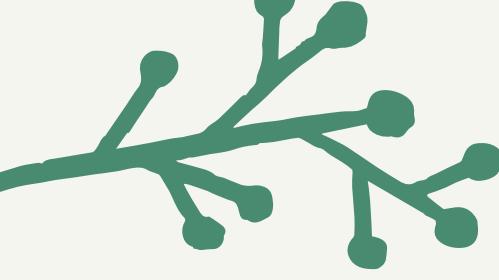
$$dW_t dW_t = dt$$

Therefore after combining the SDE equation with it, we get:

$$df(t, X_t) = \left(\frac{\partial f}{\partial t} + a(t, X) \frac{\partial f}{\partial x} + \frac{1}{2} b^2(t, X) \frac{\partial^2 f}{\partial x^2} \right) dt + b(t, X) \frac{\partial f}{\partial x} dW_t$$

Application:
Solve Black-Scholes DE





Method 2: Euler-Maruyama Method

Euler-Maruyama Method

Given the SDE initial value problem

$$\begin{cases} dX(t) = a(t, X)dt + b(t, X)dW_t \\ X(c) = X_c \end{cases}$$

we compute the approximate solution as follows:

$$\omega_0 = X_0$$

$$\omega_{i+1} = \omega_i + a(t_i, \omega_i)\Delta t_{i+1} + b(t_i, \omega_i)\Delta W_{i+1}$$

where

$$\Delta t_{i+1} = t_{i+1} - t_i$$

$$\Delta W_{i+1} = W(t_{i+1}) - W(t_i)$$

Special case: OU process

Ornstein-Uhlenbeck process

The OU process is a specific type of SDE that models mean-reverting behavior. It describes a stochastic process that tends to drift towards its long-term mean over time.

Consider the **Langevin equation**:

$$dX(t) = -\mu X(t)dt + \sigma dW_t$$

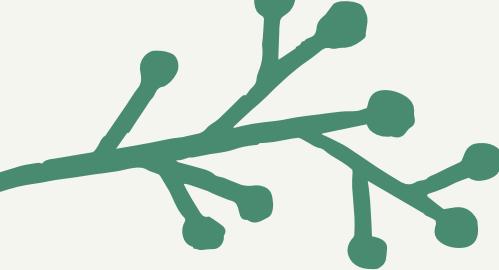
where μ and σ are positive constants. The solution of the Langevin equation is a stochastic process called the **Ornstein-Uhlenbeck process**. It was generated from an Euler-Maruyama approximation, using the steps:

$$\omega_0 = X_0$$

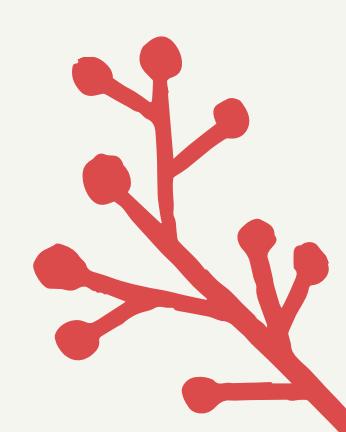
$$\omega_{i+1} = \omega_i - \mu\omega_i\Delta t_i + \sigma\Delta W_i$$

for $i = 1, \dots, n$.





Ornstein-Uhlenbeck Process



OU process: Definition

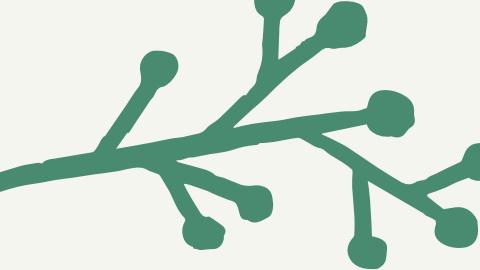
The Ornstein-Uhlenbeck process is a stochastic process that satisfies the following SDE:

$$dX_t = \kappa(\theta - X_t)dt + \sigma dW_t$$

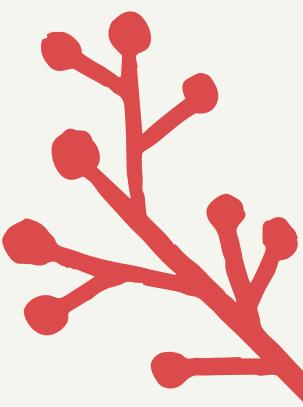
where W_t is a standard Brownian motion on $t \in [0, \infty)$. The constant parameters are:

- ▶ $\kappa > 0$ is the rate of mean reversion;
- ▶ θ is the long-term mean of the process;
- ▶ $\sigma > 0$ is the volatility or average magnitude, per square-root time, of the random fluctuations that are modeled as Brownian motions.





Ornstein-Uhlenbeck Process



OU process: Mean-reverting property

If we ignore the random fluctuations in the process due to dW_t , then we see that X_t has an overall drift towards a mean value θ . The process X_t reverts to this mean exponentially, at rate κ , with a magnitude in direct proportion to the distance between the current value of X_t and θ .

For any fixed s and t , the random variable X_t , conditional upon X_s , is normally distributed with:

$$\text{mean} = \theta + (X_s - \theta)e^{-\kappa(t-s)}$$

$$\text{variance} = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa(t-s)})$$

Observe that the mean of X_t is exactly the value derived heuristically in the solution of the ODE. The Ornstein-Uhlenbeck process is a time-homogeneous Itô diffusion.

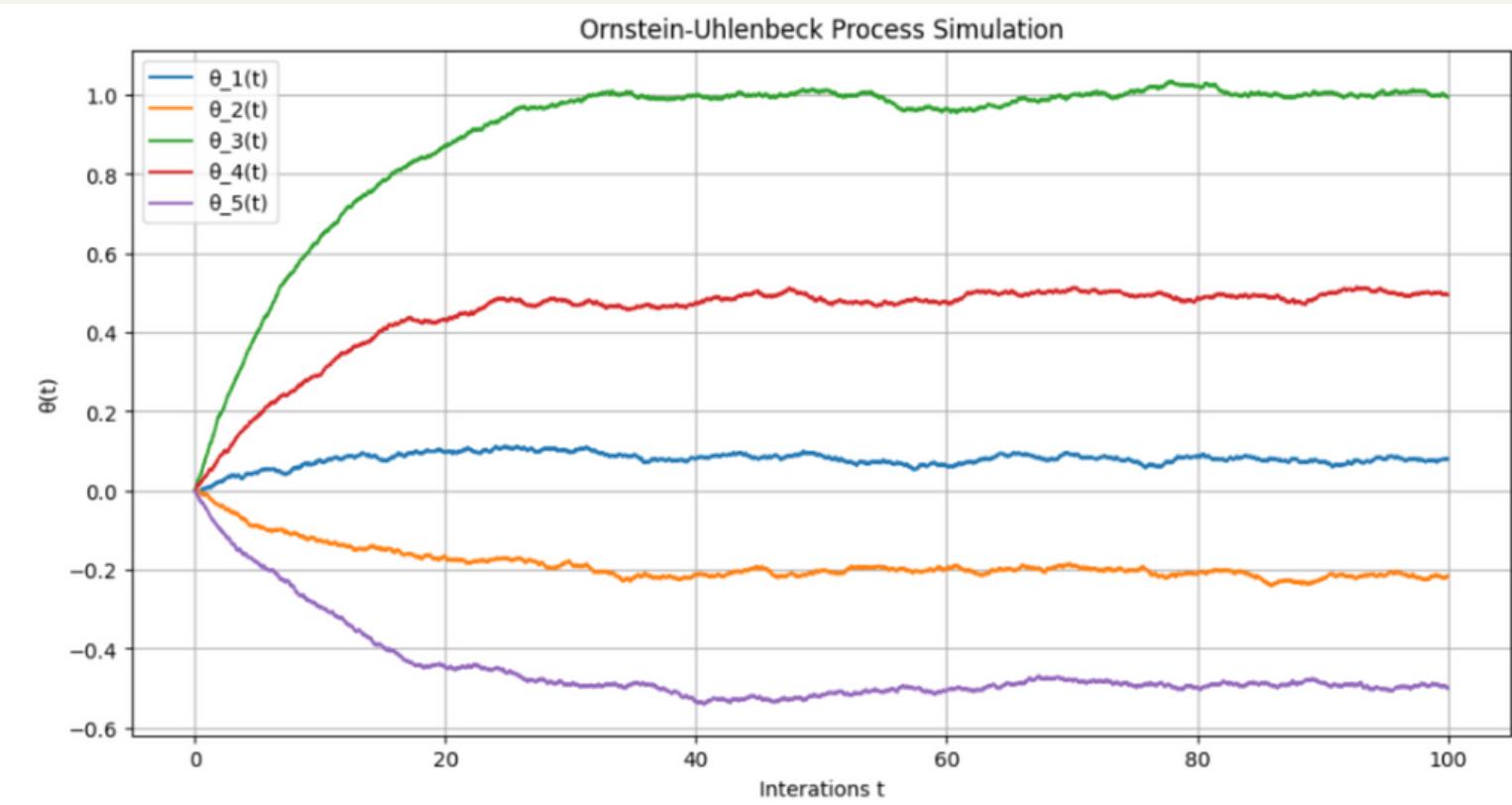


Figure 16: Simulation of OU process



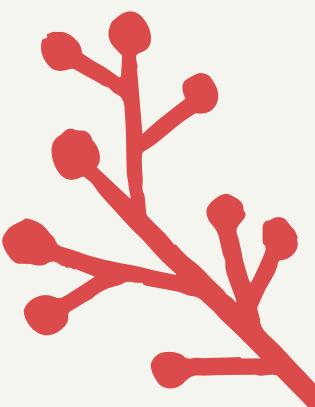
-3-

Stochastic Gradient Descent(SGD)

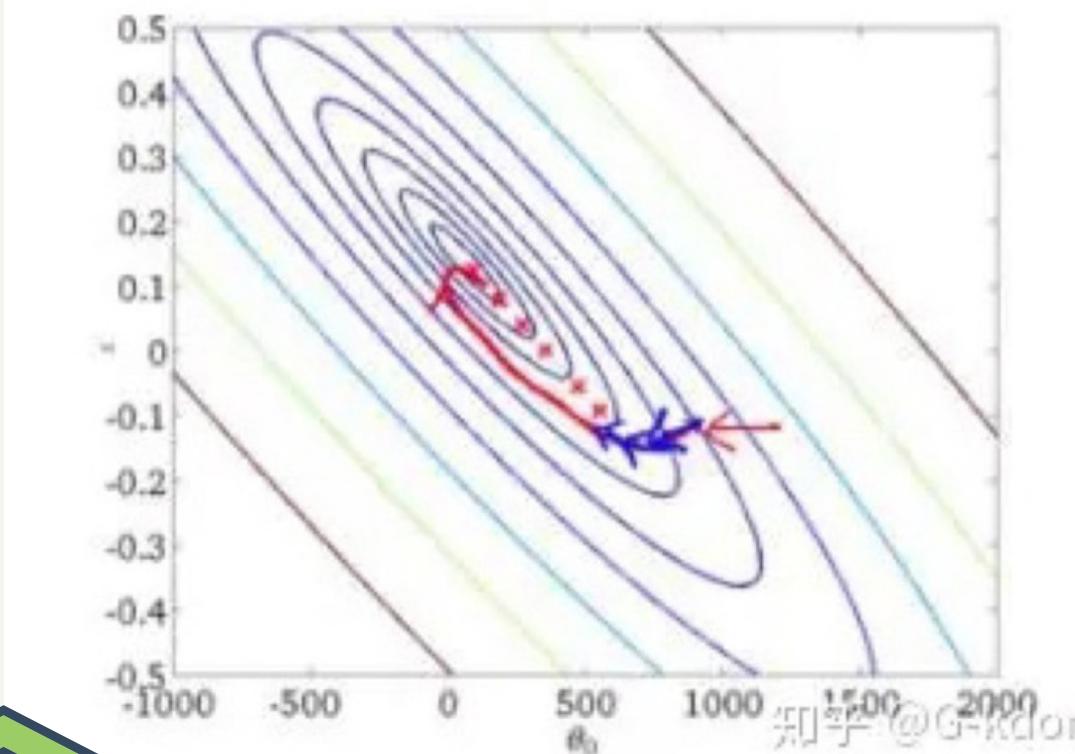




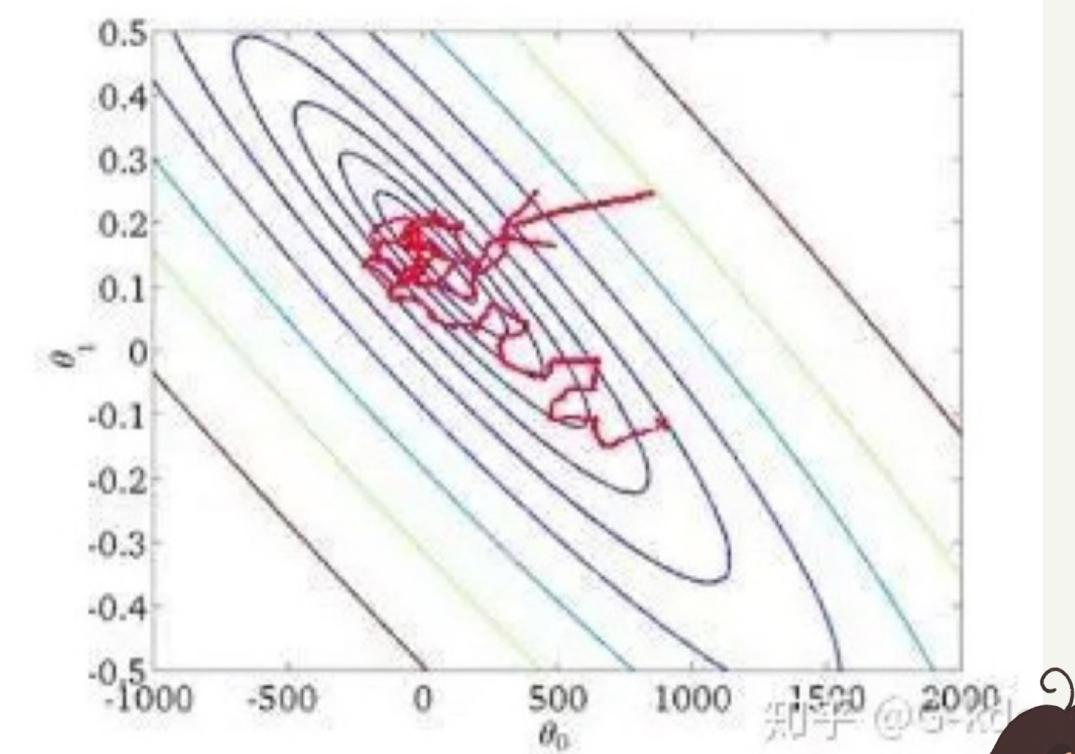
The advantage of SGD vs Batch



The **stochastic gradient descent (SGD)** aims at minimizing a function through unbiased estimates of its gradient. It is an optimization algorithm used primarily for training **large-scale** machine learning models. It's a variant of gradient descent, where instead of computing the gradient of the cost function using the entire dataset (as in batch gradient descent), it computes the gradient using a small batch of samples.



Batch

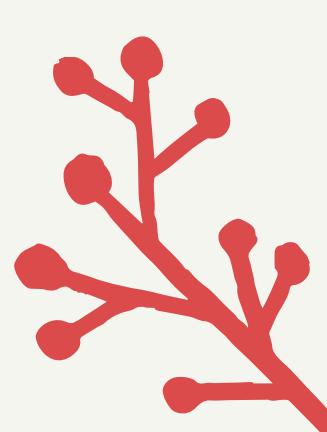


Stochastic





The Algorithm of SGD



Stochastic Gradient Descent

Solving EMR using the **standard gradient descent(GD)** on x gives the iteration scheme. First, define the gradient of f as for all $x = (x_1, \dots, x_d) \in \mathbb{R}^d$, for all $i = 1 \dots d$,

$$\nabla f(x) = \begin{pmatrix} \partial_{x_1} f(x_1, \dots, x_d) \\ \vdots \\ \partial_{x_d} f(x_1, \dots, x_d) \end{pmatrix} \in \mathbb{R}^d,$$

Then we have the recursion

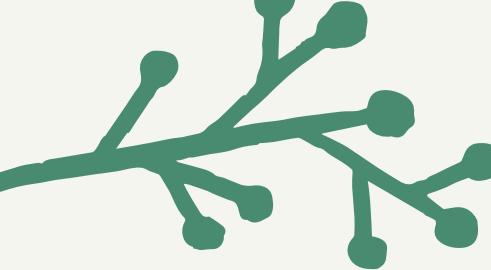
$$x_{k+1} = x_k - \eta \nabla f(x_k) = x_k - \eta \nabla \mathbb{E}_\gamma [f_\gamma(x_k)]$$

for $k \geq 0$ and η is a small step-size known as the **learning rate**.
Simple form:

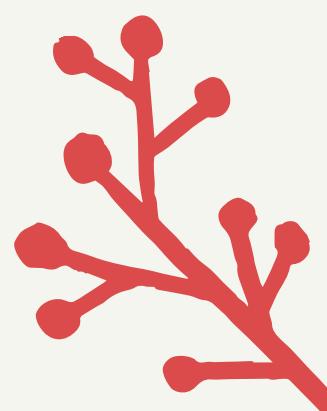
$$x_{k+1} = x_k - \eta \nabla f_{\gamma_k}(x_k)$$

where each γ_k is an i.i.d random variable with the same distribution as γ . We then have $\mathbb{E}[\nabla f_{\gamma_k}(x_k)|(x_k)] = \nabla \mathbb{E}f(x_k)$.





SGD Simulation



We use the equation of $\theta_{t+1} = \theta_t - \gamma x_t (\langle \theta_t, x_t \rangle - y_t)$ to write a Python code that simulates the SGD dynamics until time $t = 1000$, with step-size $\gamma = 0.01$, initialization $\theta_0 = \mathbf{0}$, the zero vector, $\theta^* = [0.1, -0.2, 1, 0.5, -0.5]$ and $\sigma = 0.1$. We display the test error curve upon time $\|\theta_t - \theta^*\|^2$ for several runs of the dynamics (meaning different data), and also display the two first coordinates of $(\theta_t)_t$ as well as the ones of θ^* .

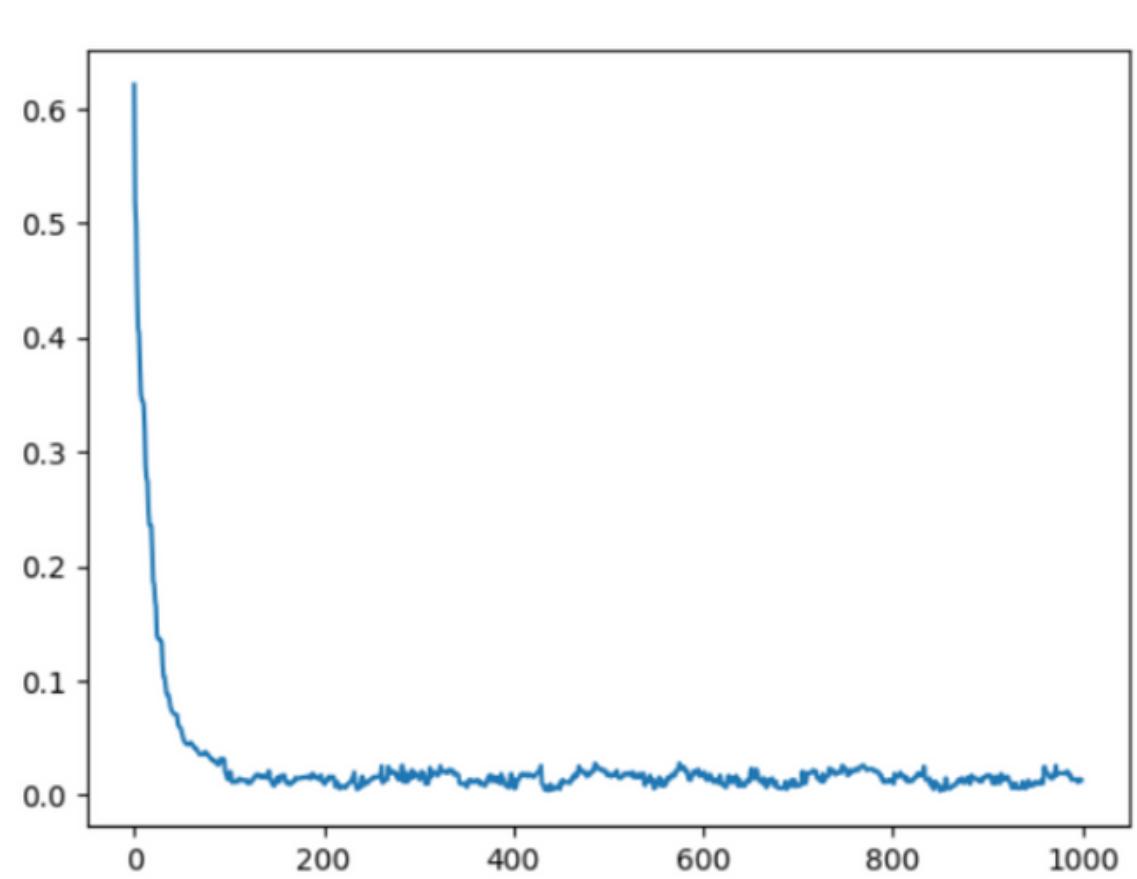


Figure 1: Simulation of test error curve

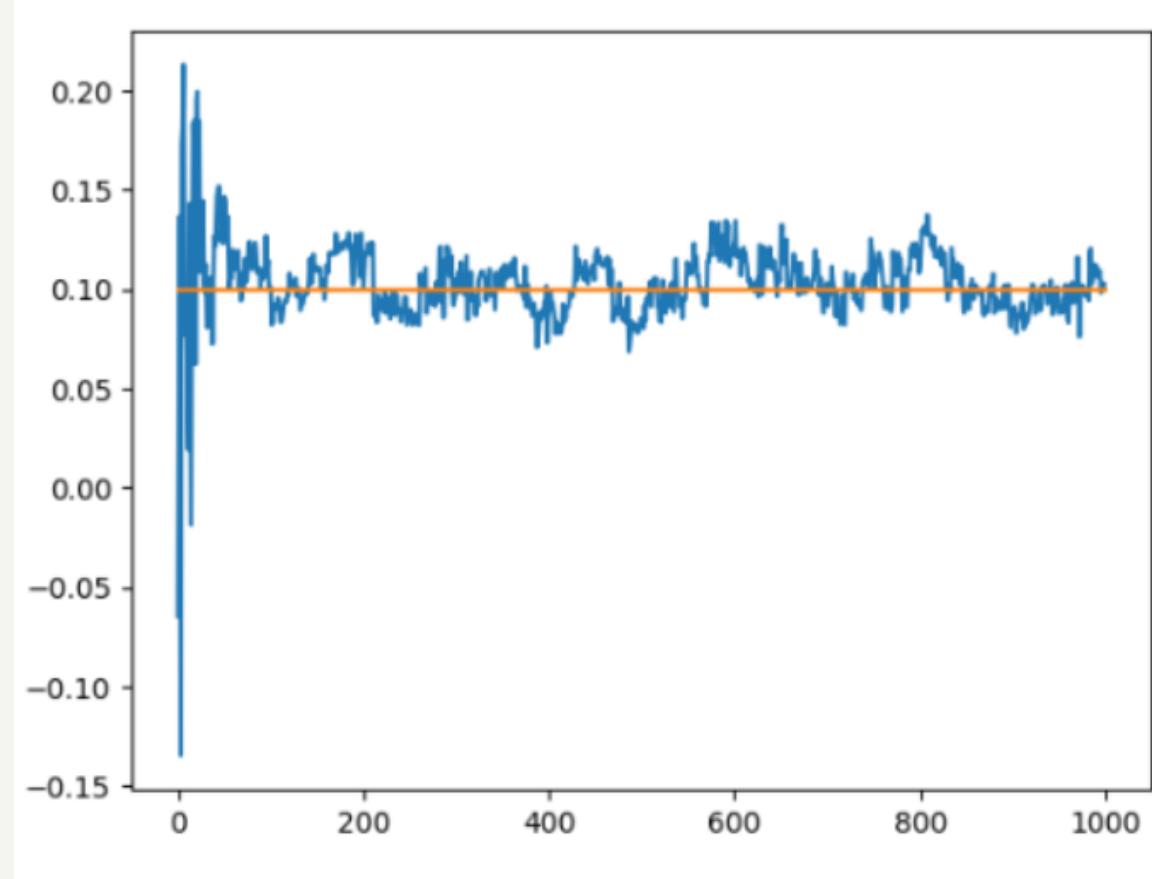


Figure 2: Simulation of θ_1

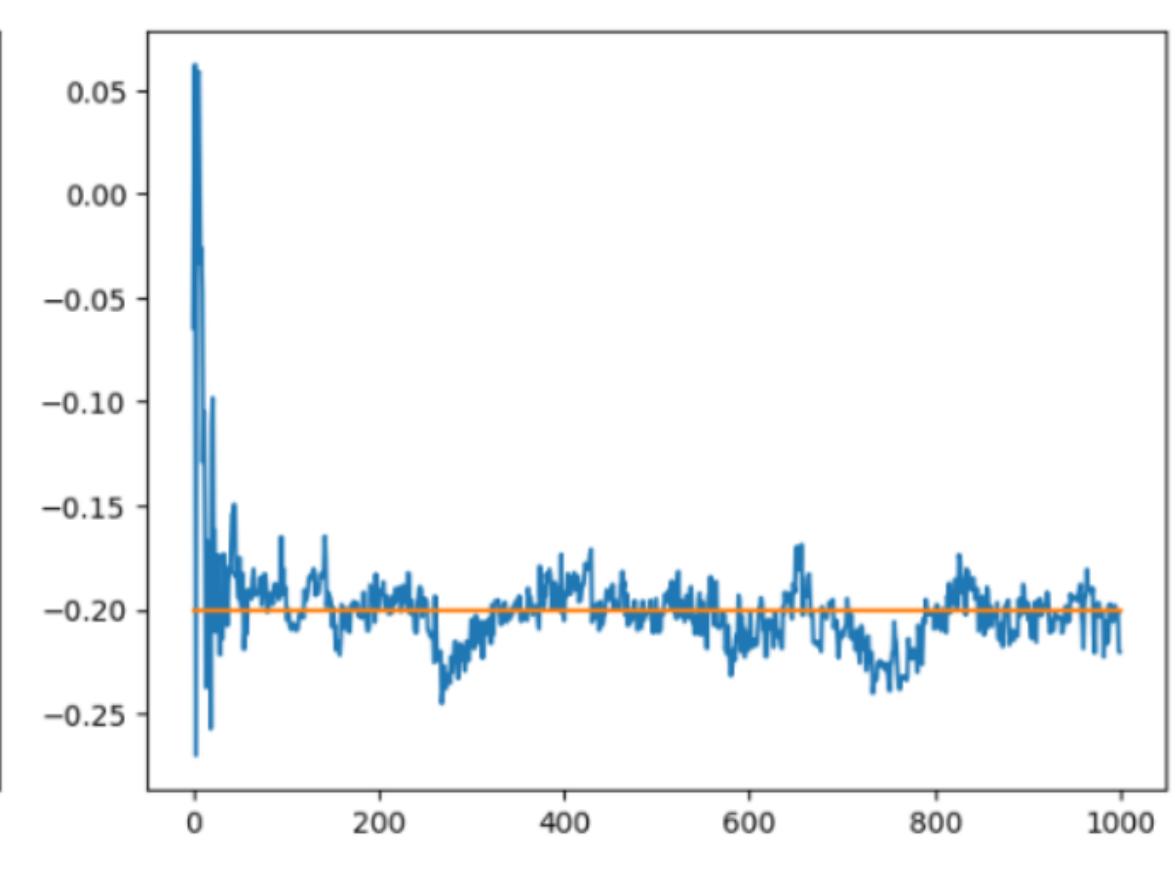
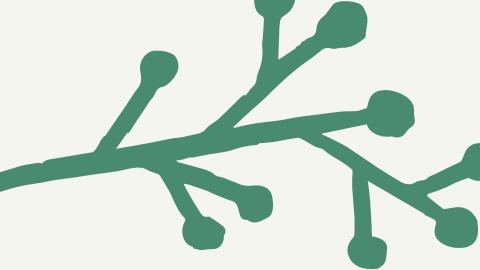


Figure 3: Simulation of θ_2



SGD Simulation: variance increases

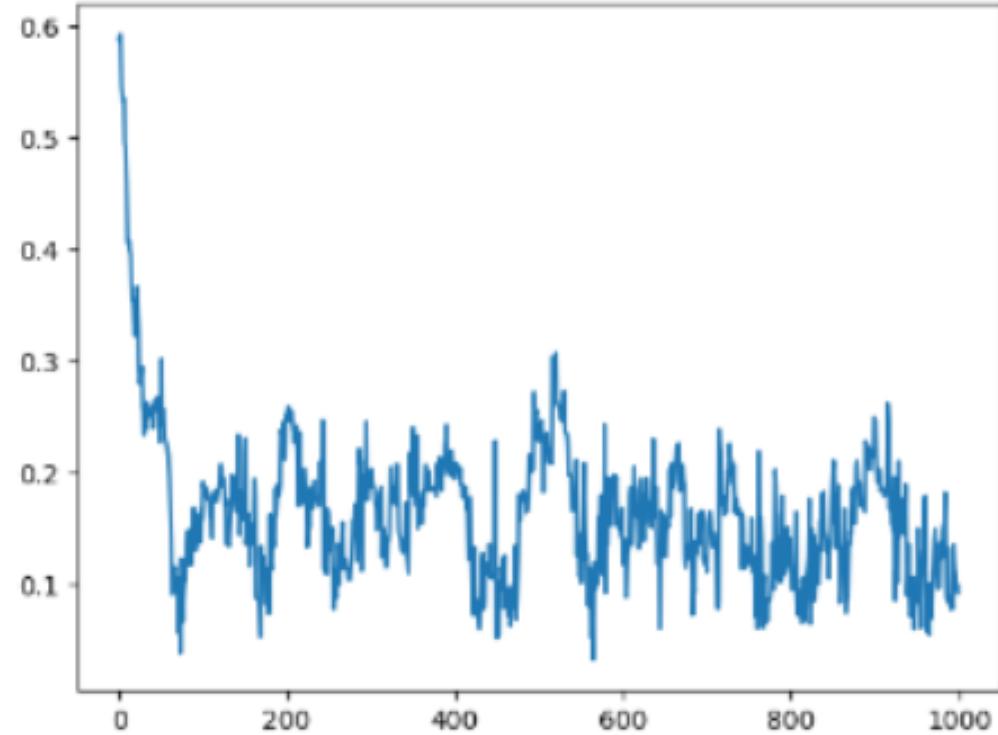
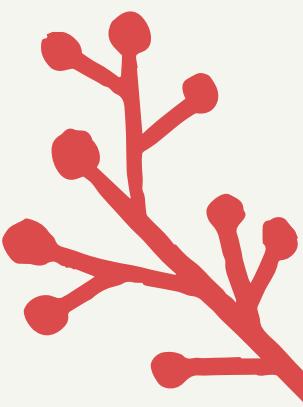


Figure 4: Test Error
when $\sigma = 1$

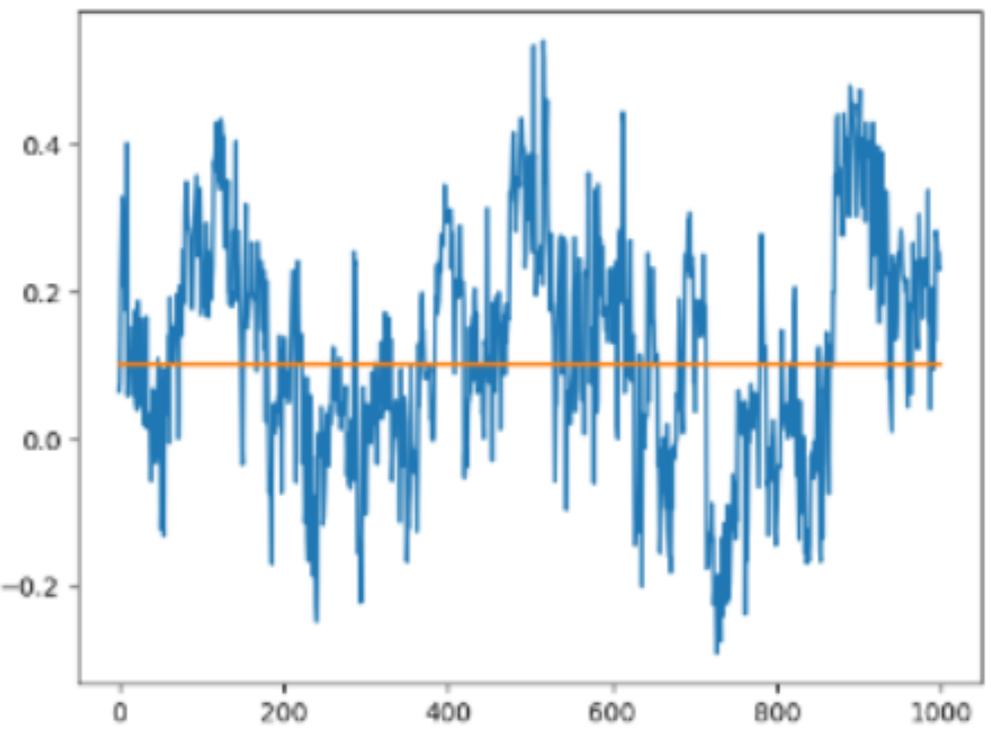


Figure 5: Simulation of θ_1
when $\sigma = 1$

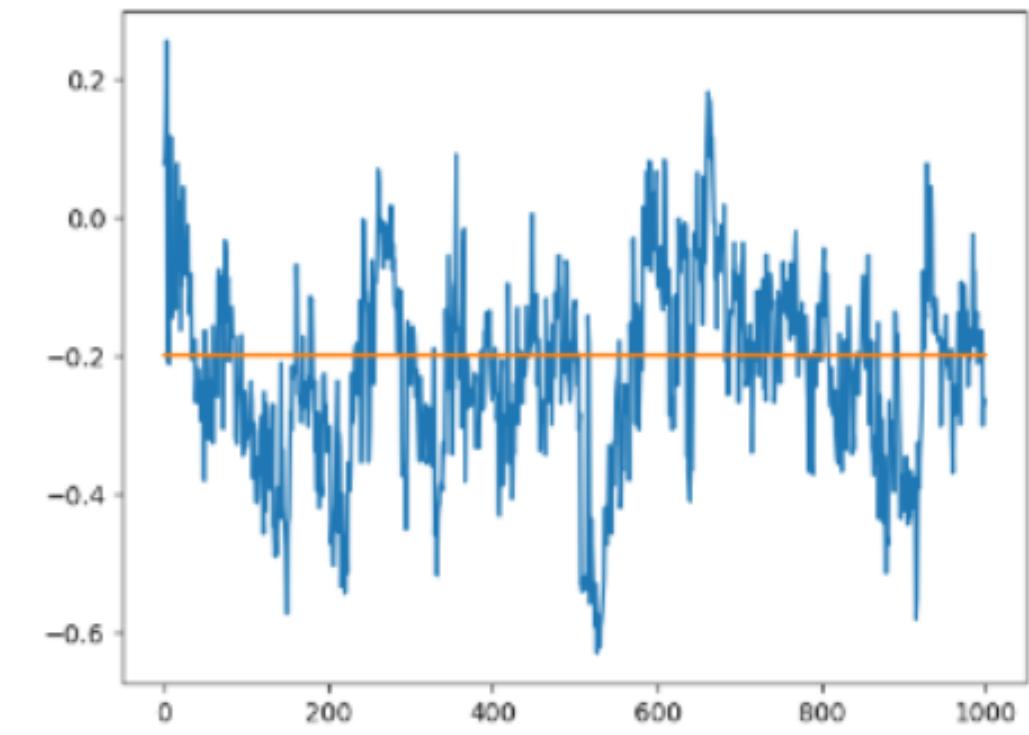
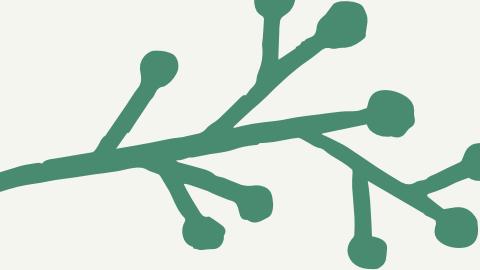


Figure 6: Simulation of θ_2
when $\sigma = 1$



SGD Simulation: step-size increases

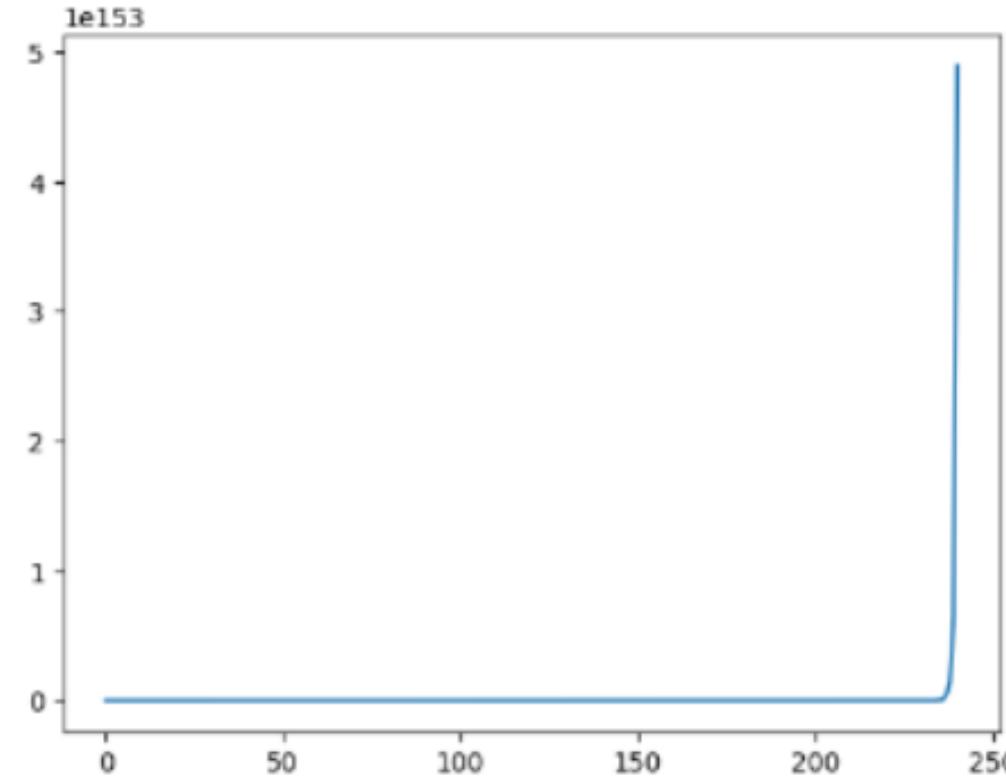
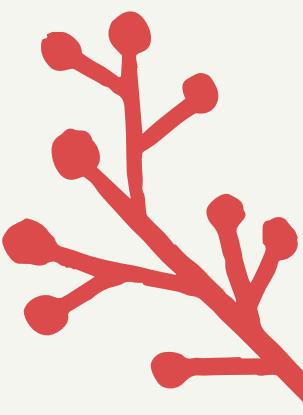


Figure 7: Test Error when
step-size bigger

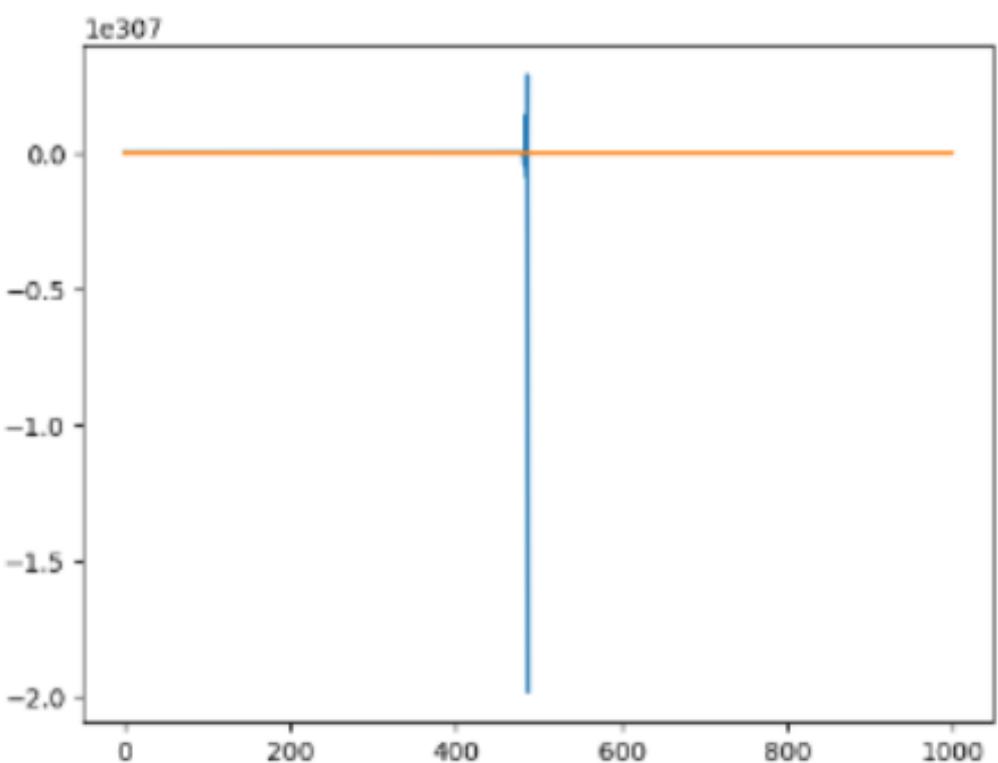


Figure 8: θ_1 when
step-size bigger

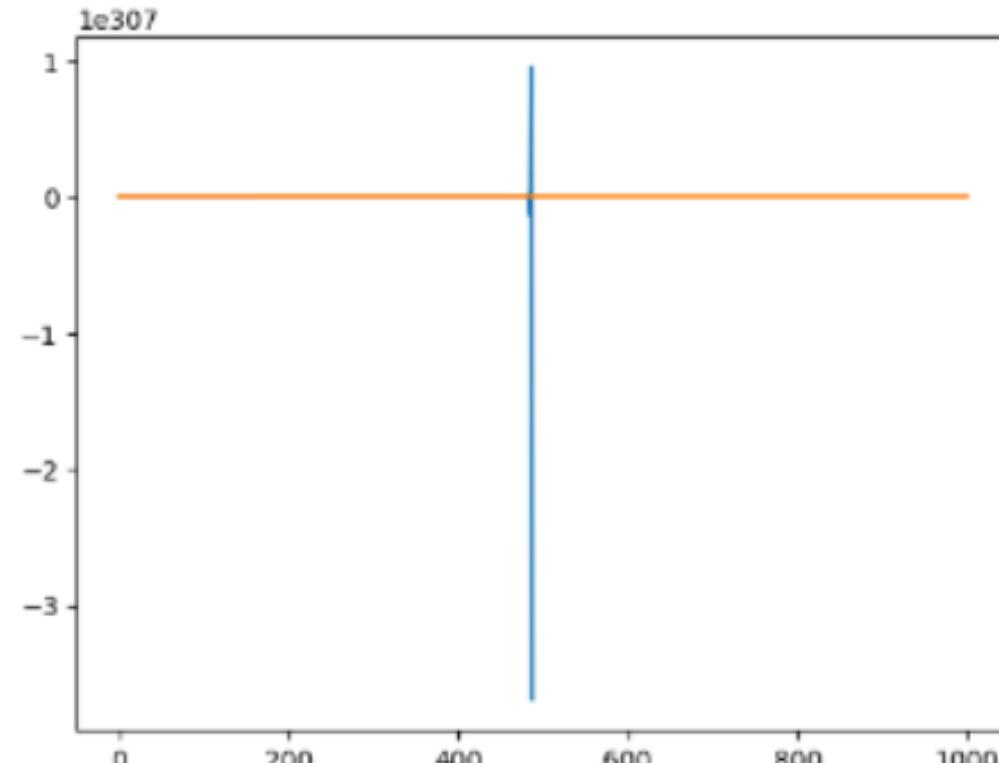
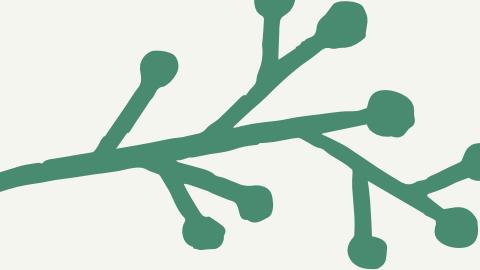


Figure 9: θ_2 when
step-size bigger



SGD Simulation: step-size decreases

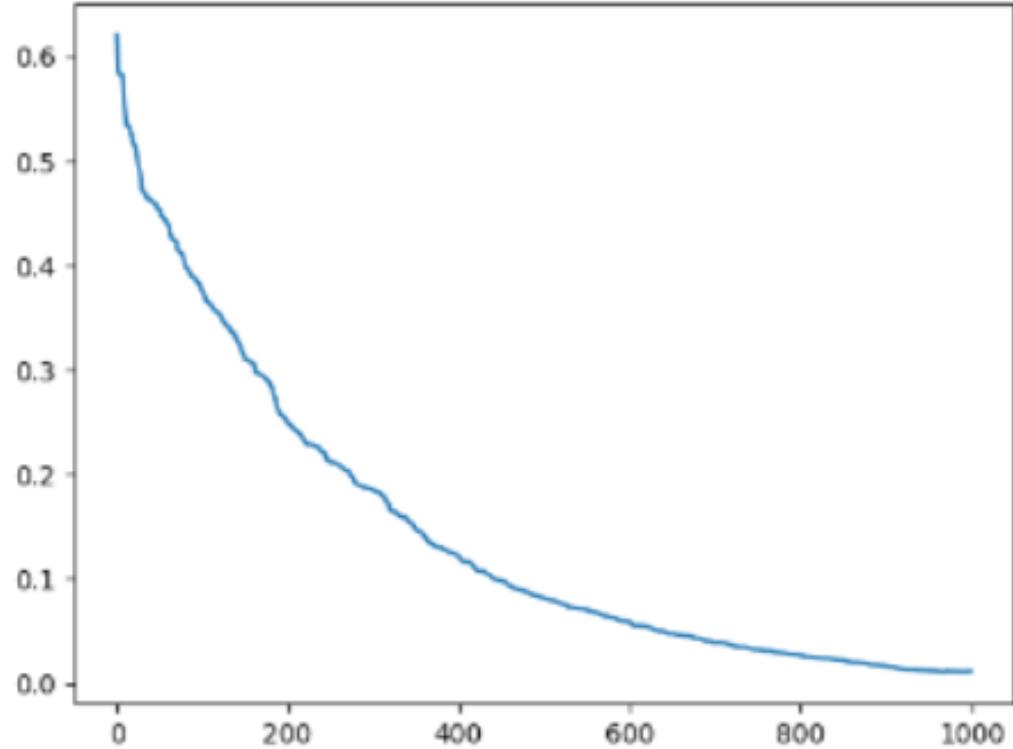
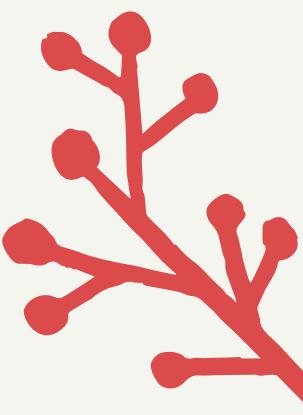


Figure 10: Test Error when
step-size smaller

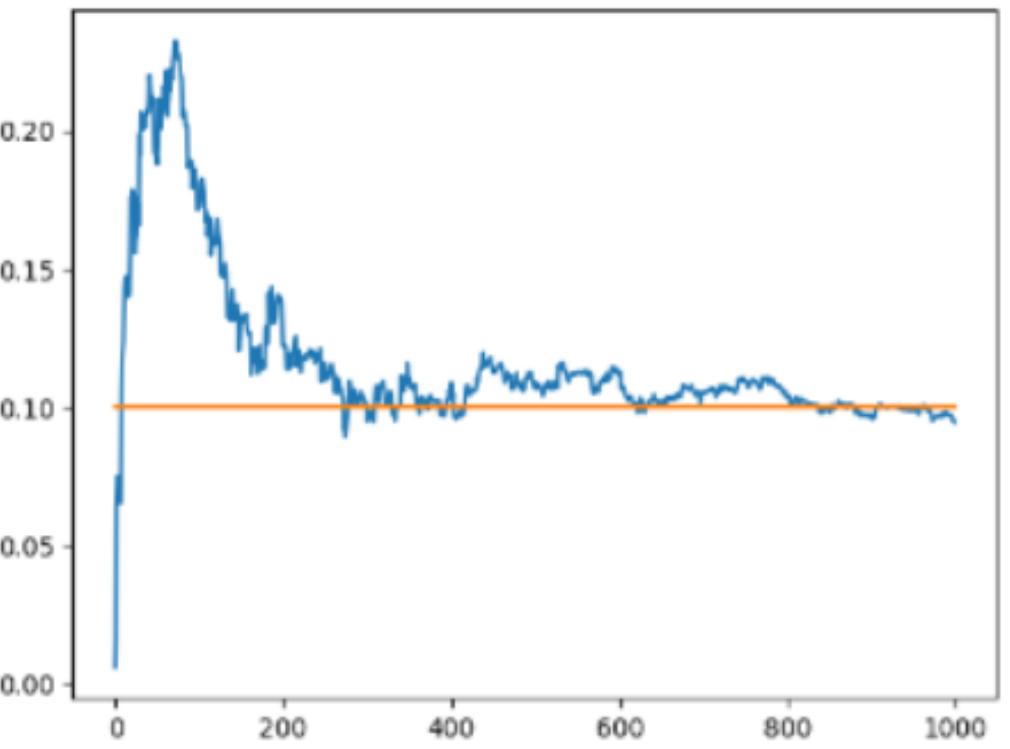


Figure 11: θ_1 when
step-size smaller

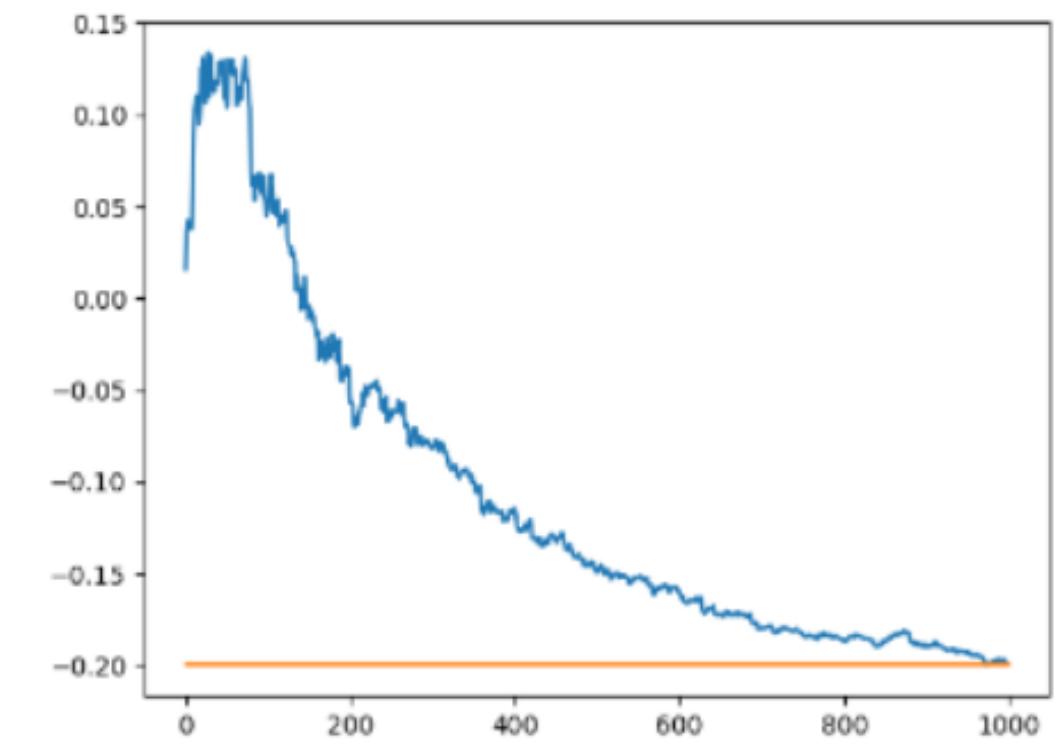


Figure 12: θ_2 when
step-size smaller

SGD Simulation: step-size depends on iteration

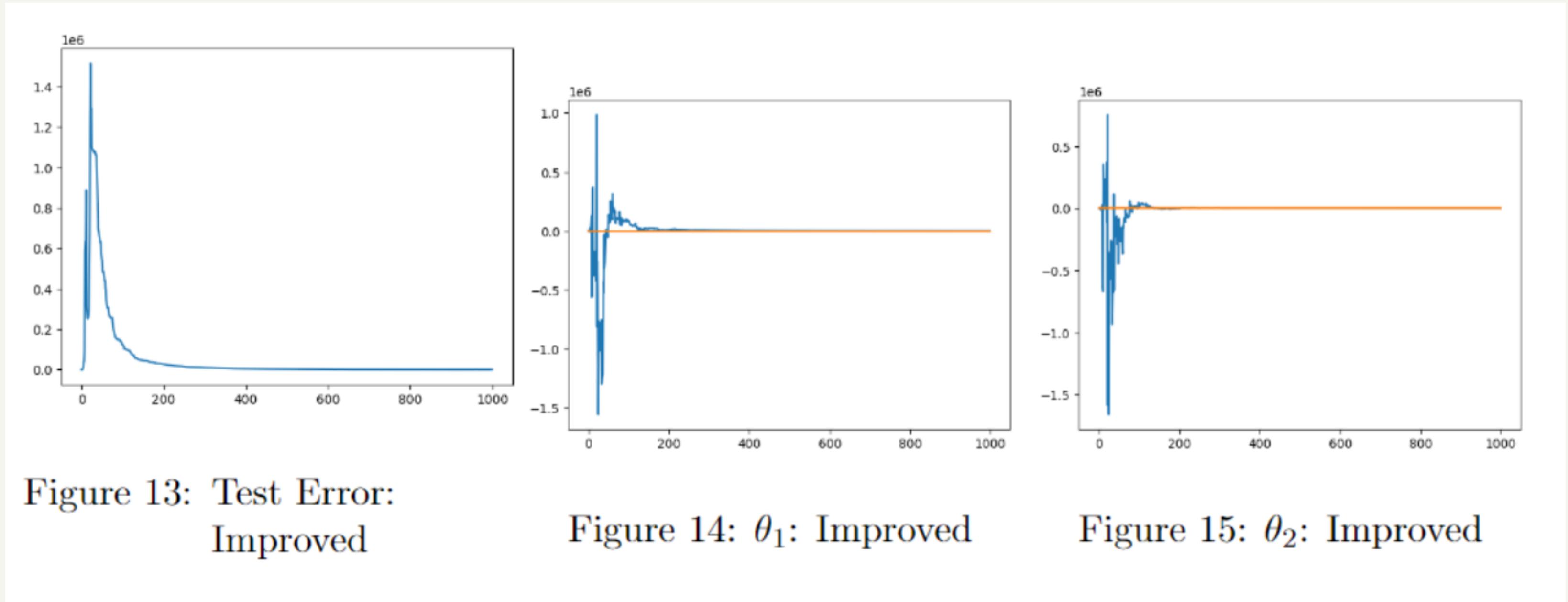


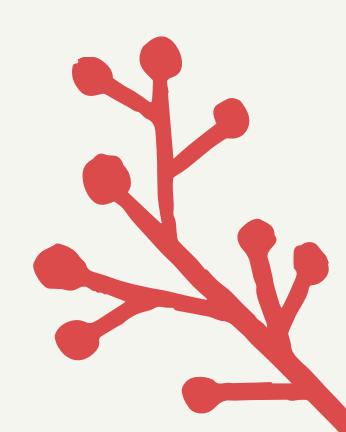
Figure 13: Test Error:
Improved

Figure 14: θ_1 : Improved

Figure 15: θ_2 : Improved



What SDE model fits well with the SGD dynamics?



Answer: Stochastic Modified Equations(SME)

match
parameters

General solution of SDE:

$$\text{.....} \rightarrow d\theta_t = b(t, \theta_t)dt + \sigma(t, \theta_t)dB_t,$$

If we apply the Euler-Maruyama discretization with step-size γ , approximating $X_{k\gamma}$ by \hat{X}_k , we obtain the following discrete iteration:

$$\hat{\theta}_{t+1} - \hat{\theta}_t = \gamma b(t, \hat{\theta}_t) + \sqrt{\gamma} \sigma(t, \hat{\theta}_t) Z_k$$

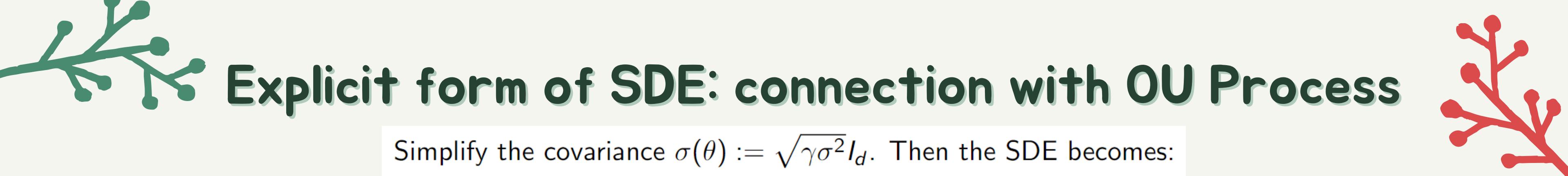
where $Z_k := B_{(k+1)\gamma} - B_{k\gamma}$ are d-dimensional i.i.d standard normal random variables. Stochastic Modified Equation:

$$\theta_{t+1} - \theta_t = -\gamma \nabla L(\theta_t) + \gamma (\nabla L(\theta_t) - \nabla I(\theta_t))$$

Then

$$\text{.....} \rightarrow d\theta_t = -\nabla L(\theta_t)dt + \sqrt{\gamma \Sigma(\theta)} dB_t$$





Explicit form of SDE: connection with OU Process

Simplify the covariance $\sigma(\theta) := \sqrt{\gamma\sigma^2}I_d$. Then the SDE becomes:

$$d\theta_t = (-(\theta_t - \theta^*)Id)dt + (\sqrt{\gamma\sigma^2}Id)dB_t$$

Match each parameter with the OU process:

$$d\theta_t = \kappa(\theta - \theta_t)dt + \sigma dW_t$$

We get:

- $\kappa = 1$.
- $\theta = \theta^*$, the long-term mean of the process matches θ^* .
- $\sigma = \sqrt{\gamma\sigma^2}$, the volatility term matches the noise factor.

The mean of the process is $\mathbb{E}(\theta_t) = \theta^* + (\mathbb{E}(\theta_0) - \theta^*)e^{-t}$.

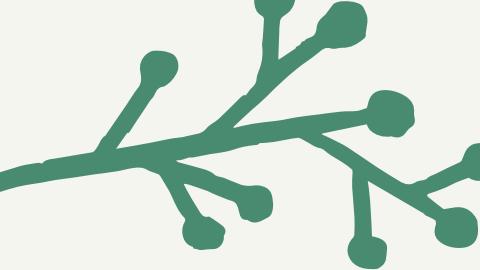
The variance of the process is $Var(X_t) = \frac{\gamma\sigma^2}{2}(1 - e^{-2t})$

The process converges to Gaussian Distribution with mean θ^* and variance $\frac{\gamma\sigma^2}{2}$, since the mean reversion term represents a force that pulls the process back towards the mean θ^* when θ_t deviates from it.

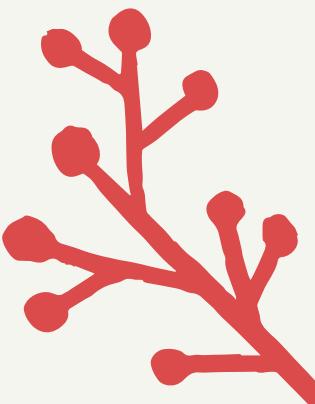
That's all for
today's
presentation!

Hope you had fun!





Reference



- Ali, A., Dobriban, E., & Tibshirani, R. (2020). The implicit regularization of stochastic gradient flow for least squares. In International Conference on Machine Learning (pp. 233–244). PMLR.
- Benaïm, M. (2006). Dynamics of stochastic approximation algorithms. In Séminaire de Probabilités XXXIII (pp. 1–68). Springer.
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. SIAM Review, 60(2), 223–311.
- Harold, J., Kushner, G., & Yin, G. (1997). Stochastic approximation and recursive algorithms and applications. Application of Mathematics, 35.
- Khasminskii, R. (2011). Stochastic stability of differential equations (Vol. 66). Springer Science & Business Media.
- Li, Q., Tai, C., & Weinan, E. (2019). Stochastic modified equations and dynamics of stochastic gradient algorithms I: Mathematical foundations. The Journal of Machine Learning Research, 20(1), 1474–1520.
- Robbins, H., & Monro, S. (1951). A stochastic approximation method. Ann. Math. Statistics, 22, 400–407.